NOV 16 2006
PATENT & TRADEMARK OFFICE

PTO/SB/21 (09-06)
Approved for use through 03/31/2007. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission

Application Number	10/046,389
Filing Date	January 16, 2002
First Named Inventor	Berger
Art Unit	2192
Examiner Name	Rutten, James D.
Attorney Docket Number	014600-0003 (B69393)

ENCLOSURES *(Check all that apply)*

- ☐ Fee Transmittal Form
- ☑ Fee Attached
- ☐ Amendment/Reply
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Reply to Missing Parts/ Incomplete Application
  - ☐ Reply to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) _____
  - ☐ Landscape Table on CD
- ☐ After Allowance Communication to TC
- ☑ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- ☐ Proprietary Information
- ☐ Status Letter
- ☑ Other Enclosure(s) (please Identify below):

Check in the amount of $500.00; return receipt postcard

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Jackson Walker L.L.P.		
Signature			
Printed name	Christopher J. Rourk		
Date	November 13, 2006	Reg. No.	39,348

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature			
Typed or printed name	Joan B. Farragher	Date	November 13, 2006

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

OIPE
NOV 16 2006
PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

Appl. No.	:	10/046,389	Confirmation No. 1678
Applicants	:	Berger et al.	
Filed	:	January 16, 2002	
Art Unit	:	2192	
Examiner	:	Rutten, James D.	
Docket No.	:	014600-0003 (B69393)	
Customer No.	:	33649	

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Certificate of Mailing Under 37 C.F.R. 1.8(a)

I hereby certify that these documents are being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to: Mail Stop Appeal Brief – Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450 on November 13, 2006.

Joan B. Farragher

ATTENTION: Board of Patent Appeals and Interferences

APPELLANT'S BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on September 7, 2006 (received by the U.S. Patent and Trademark Office on September 11, 2006) and the Final Office Action mailed June 7, 2006.

The fees required under § 1.17(c), and any required petition for extension of time for filing this brief and related fees are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

11/16/2006 MBIZUNES 00000054 10046389
01 FC:1402 500.00 OP

OIPE
NOV 16 2006
PATENT & TRADEMARK OFFICE
IAP66

TABLE OF CONTENTS

I	REAL PARTIES IN INTEREST (37 C.F.R. §41.37(c)(1))	3
II	RELATED APPEALS AND INTERFERENCES (37 C.F.R. §41.37(c)(2))	4
III	STATUS OF CLAIMS (37 C.F.R. §41.37(c)(3))	5
IV	STATUS OF AMENDMENTS (37 C.F.R. 41.37(c)(4))	6
V	SUMMARY OF THE CLAIMED SUBJECT MATTER (37 C.F.R. 41.37(c)(5))	7
VI	GROUNDS OF REJECTION TO BE REVIEWED UPON APPEAL (37 C.F.R. §41.37(c)(6))	11
VII	ARGUMENTS (37 C.F.R. 41.37(c)(7))	12
VIII	APPENDIX OF CLAIMS (37 C.F.R. 41.37(c)(8))	22
IX	EVIDENCE APPENDIX (37 C.F.R. 41.37(c)(9))	26
X	RELATED PROCEEDINGS APPENDIX (37 C.F.R. 41.37(c)(10))	27

The final page of this brief bears the practitioner's signature.

NOV 16 2006
PATENT & TRADEMARK OFFICE

I REAL PARTIES IN INTEREST (37 C.F.R. §41.37(c)(1))

The real party in interest in this appeal is:

☒ the following party:

Prelude Systems, Inc. by Assignment recorded with the U.S. Patent and Trademark Office on April 22, 2002 at Reel 012823, Frame 0856.

II RELATED APPEALS AND INTERFERENCES

(37 C.F.R. §41.37(c)(2))

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal:

A ☒ there are no such appeals or interferences.

III STATUS OF CLAIMS (37 C.F.R. §41.37(c)(3))

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 20

B. STATUS OF ALL THE CLAIMS IN APPLICATION

Claims rejected: Claims 1-20

C. CLAIMS ON APPEAL

The claims on appeal are: Claims 1-20

IV STATUS OF AMENDMENTS (37 C.F.R. 41.37(c)(4))

No amendments have been submitted subsequent to the final rejection of the claims.

V SUMMARY OF THE CLAIMED SUBJECT MATTER (37 C.F.R. 41.37(c)(5))

Claim 1 discloses a system for generating user interface code comprising: a user interface class system (by way of example and not limitation, figures and text discussed at paragraphs 0025-0031) generating user interface class code, wherein the user interface class code includes two or more user interface features that can be selected and assembled into a user interface by a user; and a handler class system (by way of example and not limitation, figures and text discussed at paragraphs 0025-0030, 0077-77, 0080) generating handler class code, wherein the handler class code includes one or more states for each user interface feature of the user interface class; wherein the user interface class and the handler class cause the selected user interface features and associated states for the user interface features to be automatically generated when the user interface code is executed.

Claim 2 includes the system of claim 1 wherein the user interface class system (by way of example and not limitation, figures and text discussed at paragraphs 0025-0031) further comprises a developer user interface class system automatically generating a developer user interface class that operates in conjunction with the user interface code to provide modified user interface features.

Claim 3 includes the system of claim 1 further comprising a developer handler class system (by way of example and not limitation, figures and text discussed at paragraphs 0025-0030, 0077-77, 0080) generating a developer handler class that operates in conjunction with the user interface code to automatically provide modified user interface states.

Claim 4 includes the system of claim 2 wherein the user interface class system (by way of example and not limitation, figures and text discussed at paragraphs 0025-0031) further comprises a site-specific user interface class system (by way of example and not limitation, figures and text discussed at paragraphs 0018, 0020, 0030, and 0033) generating a site-specific user interface class that operates in conjunction with the user interface code and the developer user interface class to provide site-specific user interface features.

Claim 5 includes the system of claim 3 wherein the developer handler class system (by way of example and not limitation, figures and text discussed at paragraphs 0025-0030, 0077-77, 0080) further comprises a site-specific handler class system (by way of example and not limitation, figures and text discussed at paragraphs 0018, 0020, 0030, and 0033) generating a

site-specific handler class that operates in conjunction with the user interface code and the developer handler class to provide site-specific user interface states.

Claim 6 includes the system of claim 1 wherein the user interface class (by way of example and not limitation, figures and text discussed at paragraphs 0025-0031) includes an accounts payable user interface class, an accounts receivable user interface class, a general ledger user interface class, a global user interface class, an inventory control user interface class, an order common user interface class, a purchase order user interface class, a report user interface class, a shipping order user interface class, a utility user interface class, a report user interface class, a test user interface class, and a template user interface class (by way of example and not limitation, figures and text discussed at paragraphs 0041-0075).

Claim 7 includes the system of claim 1 wherein the handler class (by way of example and not limitation, figures and text discussed at paragraphs 0025-0030, 0077-77, 0080) includes an accounts payable handler class, an accounts receivable handler class, a general ledger handler class, a global handler class, an inventory control handler class, an order common handler class, a purchase order handler class, a report handler class, a shipping order handler class, a utility handler class, a report handler class, a test handler class, and a template handler class (by way of example and not limitation, figures and text discussed at paragraphs 0041-0075).

Claim 8 includes a method for generating user interface code comprising: receiving a selection of user interface feature for a user interface class (by way of example and not limitation, figures and text discussed at paragraphs 0025-0031); automatically retrieving a handler associated with the user interface feature that includes one or more states; and automatically generating one or more code elements that cause a user display to be generated when executed that includes the user interface feature having the one or more states (by way of example and not limitation, figures and text discussed at paragraphs 0041-0075).

Claim 9 includes the method of claim 8 further comprising: receiving a selection of a developer user interface feature from a developer user interface class (by way of example and not limitation, figures and text discussed at paragraphs 0025-0031); and automatically generating one or more code elements that cause the user display to be generated that includes the developer user interface feature.

Claim 10 includes the method of claim 8 further comprising: receiving a selection of a developer user interface state from a developer handler class; and automatically generating one

or more code elements that cause the user display to be generated that includes the developer user interface state.

Claim 11 includes the method of claim 9 further comprising: receiving a selection of a site-specific user interface feature from a site-specific user interface class (by way of example and not limitation, figures and text discussed at paragraphs 0025-0031); and automatically generating one or more code elements that cause the user display to be generated that includes the site-specific user interface feature.

Claim 12 includes the method of claim 8 further comprising: receiving a selection of site-specific user interface state from a site-specific handler class (by way of example and not limitation, figures and text discussed at paragraphs 0025-0030, 0077-77, 0080); and automatically generating one or more code elements that cause the user display to be generated that includes the site-specific user interface state (by way of example and not limitation, figures and text discussed at paragraphs 0018, 0020, 0030, and 0033).

Claim 13 includes the method of claim 8 wherein receiving a selection of the user interface feature from the user interface class includes receiving a selection from an account payable user interface class, an accounts receivable user interface class, a general ledger user interface class, a global user interface class, an inventory control user interface class, an order common user interface class, a purchase order user interface class, a report user interface class, a shipping order user interface class, a utility user interface class, a report user interface class, a test user interface class, and a template user interface class (by way of example and not limitation, figures and text discussed at paragraphs 0041-0075).

Claim 14 includes the method of claim 8 wherein retrieving the handler associated with the user interface feature that includes one or more states includes retrieving an accounts payable handler class, an accounts receivable handler class, a general ledger handler class, a global handler class, an inventory control handler class, an order common handler class, a purchase order handler class, a report handler class, a shipping order handler class, a utility handler class, a report handler class, a test handler class, and a template handler class (by way of example and not limitation, figures and text discussed at paragraphs 0041-0075).

Claim 15 includes a system for generating software code comprising: a primary code generator receiving one or more user selections for one or more classes and generating primary software code; a developer code generator receiving the primary software code and one or more

user selections from one or more developer classes and automatically generating developer software code; a primary code editor automatically modifying one or more of the classes; and wherein the modifications made to the one or more of the classes by the primary code editor result in the generation of code that is backwards and forwards compatible with other tiers in a multi-tier architecture of the developer software code.

Claim 16 includes the system of claim 15 further comprising a site-specific code generator (by way of example and not limitation, figures and text discussed at paragraphs 0018, 0020, 0030, and 0033) receiving the primary software code, the developer software code, and one or more selections from one or more site-specific classes and generating site-specific software code, wherein the modifications made to the one or more of the classes by the primary code editor result in the generation of code that is compatible with the site-specific software code.

Claim 17 includes the system of claim 15 wherein the primary code generator further comprises a feature class including two or more features and a corresponding state class including one or more states for each feature.

Claim 18 includes the system of claim 15 wherein the developer code generator further comprises a developer feature class including two or more developer features and a corresponding developer state class including one or more developer states for each developer feature.

Claim 19 includes the system of claim 16 wherein the site-specific code generator (by way of example and not limitation, figures and text discussed at paragraphs 0018, 0020, 0030, and 0033) further comprises a site-specific feature class including two or more site-specific features and a corresponding site-specific state class including one or more site-specific states for each site-specific feature.

Claim 20 includes the system of claim 15 wherein the primary code generator further comprises a user interface class (by way of example and not limitation, figures and text discussed at paragraphs 0025-0031) including two or more user interface features and a corresponding state class including one or more states for each user interface feature.

VI GROUNDS OF REJECTION TO BE REVIEWED UPON APPEAL (37 C.F.R. §41.37(c)(6))

1. Whether claims 15-20 fail to comply with the written description requirement.

2. Whether claims 1 and 3 through 7 fail to set forth the subject matter which the Applicants regard as their invention.

3. Whether claims 1-3, 5, 8-10, 12, 15, 17, 18 and 20 are anticipated by Chiang.

4. Whether claims 4, 11, 16, and 19 are obvious over Chiang in view of Hamilton.

5. Whether claims 6, 7, 13 and 14 are obvious over Chiang and further in view of Bickerton, in view of Wong, in view of Collins, in view of Andrews, in view of Pavela.

VII ARGUMENTS (37 C.F.R. 41.37(c)(7))

1. Claims 15-20 comply with the written description requirement.

In the Office action mailed June 7, 2006, the Examiner states at paragraph 11 that claims 15-20 "contain subject matter which was not described in the specification in such a way to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention." In particular, as to claim 15, the Examiner asserts in paragraph 12 that "no support could be found for a primary code editor that automatically modifies classes. It is not evident that an editor would be able to automatically modify classes." Nevertheless, the Examiner fails to address why the disclosed code generation modules that create software code that generates a user interface fail to provide support for claim 15.

Claim 15 includes "a primary code generator receiving one or more user selections for one or more classes and generating primary software code; a developer code generator receiving the primary software code and one or more user selections from one or more developer classes and automatically generating developer software code; a primary code editor automatically modifying one or more of the classes." The primary code generator is disclosed at paragraph [0020] and includes "code generation modules that create software code that generates a user interface." One of these code generation modules is the user interface class system 110 that "can include one or more user interface classes, such as data input classes, report generation classes, order processing classes, or other suitable classes" that "each have a corresponding set of available states." Although automatic generation of code by the primary code generator is not claimed, automatic code generation is disclosed, as evidenced by the fact that the Examiner does not also reject the claimed developer code generator "**automatically** generating developer software code" under 35 U.S.C. 112, first paragraph (which is disclosed at paragraph [0026], which states that a "developer can then use developer code generator 106 to modify [i.e., edit] the code generated by primary code generator 106," and at paragraph [0027] which states that "the code generated through developer code generator 106 is compatible with the new code generated by primary code generator 104.") Thus, automatic generation of code and subsequent editing [i.e., modification] and automatic generation of the edited code is clearly disclosed, supported and enabled, and the Examiner does not indicate otherwise.

Furthermore, as disclosed at paragraph [0086], "the developer code can be generated by first displaying the primary code, such as using a primary code editor that identifies components, options, access and utilization, states, or other suitable data, and that further allows a user to modify the primary code, add new features, components, options, or states, modify existing features, components, options, or states, or otherwise change the functionality of the primary code." One of ordinary skill would understand that the primary code generator disclosed at paragraph [0020] that includes "code generation modules that create software code that generates a user interface" could operate with the disclosed primary code editor that identifies components, options, access and utilization, states, or other suitable data, and that further allows a user to modify the primary code, add new features, components, options, or states, modify existing features, components, options, or states, or otherwise change the functionality of the primary code, and that the primary code generator that automatically generates the primary code would be able to automatically generate the code as modified by the primary code editor. In any event, the Examiner has failed to provide any discussion of why this disclosure fails to enable the claimed invention, and the rejection of claim 15 must be REVERSED.

Claims 16-20 were only rejected as being dependent on claim 15, but they in fact are original claims and provide further enablement for the claimed primary code editor. Consider claim 16, which includes a site-specific code generator receiving the primary software code, the developer software code, and one or more selections from one or more site-specific classes and generating site-specific software code, wherein the modifications made to the one or more of the classes by the primary code editor result in the generation of code that is compatible with the site-specific software code. The process disclosed and claimed in claim 16 is an automatic code generation process using the primary code editor. The Examiner has failed to explain why that claim or the others, which were not rejected and which therefore must be presumed to be enabled, fail to also provide enablement for the claimed primary code editor.

The Examiner asserts that "no support is found for a developer software code that has a multi-tier architecture," but further notes that "the developer code could be viewed as being a layer, or 'tier' in the layered architecture," and concludes that "this limitation will be interpreted as –compatible with other tiers in a multi-tier architecture that includes the developer software code." The Applicants do not object to that construction, which is conceded to be enabled by the Examiner.

2. **Claims 1 and 3 through 7 clearly set forth the subject matter which the Applicants regard as their invention.**

Claims 1 and 3-7 were rejected under 35 U.S.C. 112, second paragraph, on the grounds that a statement made in the response filed March 1, 2006 "indicates that the invention is different from what is defined in the claim(s)." This rejection is legally flawed, as it "is axiomatic that the claims define the invention which an applicant believes is patentable." *In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993) (citations omitted). Furthermore, to the extent that there is any tension between the claims and the statement in the response, it is because the Examiner has imported the term "elimination of manual programming" into the statements made in the response. Nothing in the statements made in the response says that all manual programming must be eliminated, only that certain manual programming steps are replaced by the claimed automatic code generation, such as to make lower layers or tiers of code compatible with new versions of higher layers or tiers or for other suitable purposes.

The Examiner further asserts that automatic generation of states and features is different from automatic generation of code, and that automatic generation of states and features is "an inherent feature of any computer generated GUI, and does not distinguish the claim from the prior art." However, the Examiner's construction of automatic generation of code as excluding automatic generation of states and features is at odds with the specification, which discloses at paragraph [0086] that "the developer code can be generated by first displaying the primary code, such as using a primary code editor that identifies components, options, access and utilization, **states**, or other suitable data, and that further allows a user to modify the primary code, add new **features**, components, options, or states, modify existing features, components, options, or states, or otherwise change the functionality of the primary code." (Emphasis added) While the Examiner may consider automatic generation of states and features to be "an inherent feature of any computer generated GUI, [that] does not distinguish the claim from the prior art," this construction is not only contrary to the definition of terms provided by the Applicant, and therefore incorrect as a matter of law, it is also a double non sequitur.

First, even if automatic generation of states and features is "an inherent feature of any computer generated GUI, [that] does not distinguish the claim from the prior art," the rejection under 35 U.S.C. 112, second paragraph, ***is not a prior art rejection***. Whether the claim meets

the requirements of 35 U.S.C. 112, second paragraph, is a different question from whether the claim meets the requirements of 35 U.S.C. 102 or 103. Thus, it is irrelevant to the rejection under 35 U.S.C. 112, second paragraph, whether the claim is distinguishable over the prior art.

Second, even if automatic generation of states and features is "an inherent feature of any computer generated GUI, [that] does not distinguish the claim from the prior art," ***that is not a claim limitation at issue***. Those claim limitations include "a user interface class system generating user interface class code, wherein the user interface class code includes two or more user interface features that can be selected and assembled into a user interface by a user; and a handler class system generating handler class code, wherein the handler class code includes one or more states for each user interface feature of the user interface class; wherein the user interface class and the handler class cause the selected user interface features and associated states for the user interface features to be automatically generated when the user interface code is executed." As a computer generated GUI does not include a user interface class system, a handler class system, or other claim elements, the Examiner's assertions regarding a computer generated GUI is irrelevant. This rejection must also be REVERSED.

3. **Chiang fails to disclose each element of claims 1-3, 5, 8-10, 12, 15, 17, 18 and 20.**

The Examiner has construed the claims to exclude automatic generation of code, as previously discussed. As such, because the Examiner has applied an improper claim construction and has relied on art that allegedly anticipates that improper claim construction to reject the claims, the Examiner has failed to demonstrate that the cited art discloses each element of claims 1-3, 5, 8-10, 12, 15, 17, 18 and 20. Claim construction is a question of law, and is reviewed *de novo*. *Markman v. Westview*, 52 F.3d 967, 34 USPQ2d 1321 (Fed. Cir. 1995), *aff'd* 116 S.Ct. 1384 (1996).

As previously discussed, Claim 1 includes "a user interface class system generating user interface class code, wherein the user interface class code includes two or more user interface features that can be selected and assembled into a user interface by a user; and a handler class system generating handler class code, wherein the handler class code includes one or more states for each user interface feature of the user interface class; wherein the user interface class and the handler class cause the selected user interface features and associated states for the user interface

features to be automatically generated when the user interface code is executed." The Examiner has construed these claim limitations to exclude automatic code generation. Thus, under the Examiner's construction, "a user interface class system generating user interface class code" does not automatically generate code, but reads on manual generation of code. This can be seen in the section of Chiang cited by the Examiner at paragraph 18 of the Office action mailed June 7, 2006, which recites "one or more graphic designers and/or business analysts 210 creating web application screens." Thus, even though the Applicants have gone out of their way to distinguish the invention from manual code generation, and have presented arguments to clarify that issue, the Examiner has concocted a claim construction that is not only used to improperly reject the claims under 35 U.S.C. 112 paragraphs 1 and 2, but also to reject the claims over ***the very manual processes that have been distinguished over***! Once the proper claim construction is applied to the claims, it can be readily seen that the manual processes of the prior art fail to anticipate the claimed invention.

Examples of the manual processes described in Chiang that have been automated can be readily found in the specification. For example, consider paragraph [0021]:

> [0021] In one exemplary embodiment, user interface class system 110 can generate a data entry class, such as a customer account data entry class. The customer account data entry class can include one or more data entry features that may be required to define a customer account, such as a customer name, customer address, customer account number, shipping address, contact name and address, and other suitable data. User interface class system 110 allows a user to select from available classes, such as the customer account entry class, and to view the available features, components and functions performed by the class. User interface class system 110 allows a user to determine the available options, how to access and utilize the options, and otherwise allows the user to design user interface screens for use in receiving data from a user and presenting data to the user.

Thus, the user interface class system 110 of primary code generator 104 eliminates the need for "one or more graphic designers and/or business analysts 210 creating web application screens," but the Examiner has imposed a claim construction simply for bringing the claims within the ambit of the prior art. This claim construction is improper, must be REVERSED, and as well as the rejection of claim 1.

The Examiner's construction of claim 2 likewise relies improperly on manual programming, even though it is acknowledged at paragraph 7 of the Office action mailed June 7,

2006 that claim 2 is "directed to automatic code generation." Figure 2, element 210, which the Examiner asserts discloses claim 2 under the Examiner's construction, is a box showing "graphic designers/business analysts." Claim 2 includes "the system of claim 1 wherein the user interface class system further comprises a developer user interface class system automatically generating a developer user interface class that operates in conjunction with the user interface code to provide modified user interface features." Thus, claim 2 includes at least two distinct elements: a developer user interface class system and a user interface class system. Figure 4, element 405 and Figure 6 of Chiang, also relied on by the Examiner, only shows a single item, for graphical user interfaces. Thus, any modified user interface features in the GUI of Chiang must be re-implemented by a developer if new user interface code is distributed by the "graphic designers/business analysts" of Chiang. Using the example from paragraph [0021] discussed above, if a developer customizes customer account data entry implemented by the graphic designers or business analysts of Chiang, any subsequent changes to the customer account data entry implemented by the graphic designers or business analysts of Chiang will require the developer to re-implement the customization, because Chiang fails to disclose the claimed a developer user interface class system and a user interface class system, as well as other limitations of claim 1 as discussed above. The rejection of claim 2 must also be REVERSED.

Claim 3 includes a developer handler class system generating a developer handler class that operates in conjunction with the user interface code to automatically provide modified user interface states. The Examiner again construes this element as web developers 215 and the single event handler 415 of Chiang, which suffer from the same problem as the other claims – changes made at a higher "layer" (which is not even disclosed by Chiang) would be incompatible with changes that were previously made at a lower "layer." More specifically, Chiang fails to disclose the claimed handler class system and developer handler class system, when those elements are properly construed. The rejection of claim 3 must also be REVERSED.

Claim 5 includes "the developer handler class system further comprises a site-specific handler class system generating a site-specific handler class that operates in conjunction with the user interface code and the developer handler class to provide site-specific user interface states." The Examiner construes this to be that which is disclosed at paragraph [0071] of Chiang, which states that "additional files 635 are generated by web application generator 205, allowing information to be initially set by web developers 215, but changeable by application

administrators without requiring a recompilation of the application source code." Nevertheless, the cited art fails to disclose the functionality that would be provided by the claimed invention, which would allows the web developers 215 to make changes to the files 635 that are compatible with any changes that were previously made by the application administrators. The site-specific handler class system of claim 5 would provide such functionality, and properly construed, Chiang fails to disclose the elements of claim 5. Furthermore, as the Examiner admits that Chiang fails to disclose site-specific user interface features in the rejection of claim 4 at paragraph 20 of the Office action mailed June 7, 2006, Chiang simply cannot provide a basis for the rejection of claim 5 under 35 U.S.C. 102 based on the Examiner's admissions regarding the teachings of the prior art. This rejection must also be REVERSED.

In regards to claim 8, the Examiner has construed the claim to omit the term "automatically," so as to read the claim on manual processing. *See* page 9 of Office action mailed June 6, 2007. It is interesting to note that the Examiner's construction is applied inconsistently – when automatic functions can be found in Chiang, the Examiner relies on those functions as basis for the rejection (even though those functions are not related to the claim limitations); when automatic functions can not be found, the Examiner decides not to give the term "automatic" meaning and relies on manual actions as a basis for rejection. This is simply improper, and must be REVERSED.

In regards to claims 9, 10, and 12, the Examiner asserts that these have been addressed in the rejection of claims 2, 3 and 5. Claim 2 was rejected over the manual actions of graphic designers/business developers, and claim 3 was rejected over the manual actions of web developers, whereas claim 9 includes "receiving a selection of a developer user interface feature from a developer user interface class; and automatically generating one or more code elements that cause the user display to be generated that includes the developer user interface feature," claim 10 includes "receiving a selection of a developer user interface state from a developer handler class; and automatically generating one or more code elements that cause the user display to be generated that includes the developer user interface state," and claim 12 includes "receiving a selection of site-specific user interface state from a site-specific handler class; and automatically generating one or more code elements that cause the user display to be generated that includes the site-specific user interface state." As discussed, the system disclosed in the specification eliminates the need for manual functions to be performed by graphic designers,

business developers and web developers, and also creates an architecture that allows changes made at higher architecture levels to be compatible with prior changes made at lower architecture levels. Thus, the claimed invention would not be anticipated by the manual activities disclosed in Chiang, as those manual activities are not compatible with prior modifications, as previously described. Furthermore, as the Examiner admits that Chiang fails to disclose site-specific user interface features in the rejection of claim 4 at paragraph 20 of the Office action mailed June 7, 2006, it simply cannot provide a basis for the rejection of claim 12 under 35 U.S.C. 102. These rejections must be REVERSED.

In regards to claim 15, the Examiner has construed the claims so as to avoid addressing the term "automatically," as previously discussed. In addition, the Examiner construes "a multi-tiered architecture" to be the system shown in Figure 4 of Chiang. Elements 405, 415 and 420 are not a multi-tiered architecture, as admitted by the Examiner in the construction adopted in paragraph 13 of the June 7, 2006 Office action. The graphical user interfaces 405, event handler 415 and business logic 420 of Chiang are at best only parts of a single tier of what would be implemented in a multi-tier architecture. As used in the specification, which the claim terms must be construed in light of, the tiers of the "multi-tiered architecture" would include one or more of the primary, developer and site-specific layers, not the graphical user interfaces 405, event handler 415 and business logic 420 of Chiang. As the Examiner acknowledges that Chiang fails to disclose site-specific user interface features in the rejection of claim 4, the rejection of claim 15 must be REVERSED.

4. **Chiang in view of Hamilton fails to provide a prima facie basis for the rejection of claims 4, 11, 16 and 19.**

Although the Examiner rejects claims 4, 11, 16 and 19 over Chiang and Myers as applied to claims 2, 9 and 15, in view of Hamilton, this appears to be a sloppy error, as the Examiner has only discussed Chiang, and does not discuss any Myers reference. As such, the rejection is interpreted to be over Chiang in view of Hamilton. In the event a new reference is applied to these claims, the Applicants reserve the right to address any such new grounds of rejection.

Chiang in view of Hamilton fails to provide a prima facie basis for the rejection of claims 4, 11, 16 and 19, as they fail to disclose each element of the claimed invention. In particular, Chiang discloses a single layer architecture, and Hamilton also discloses a single layer

architecture. However, even assuming *arguendo* that Chiang and Hamilton both disclose a two layer architecture (they do not), two plus two do not equal three (or even four in this case), because the two layers of Chiang and Hamilton as interpreted by the Examiner are the same two layers. In contrast, claim 4 discloses "a site-specific user interface class that operates in conjunction with the user interface code and the developer user interface class." Neither Chiang nor Hamilton discloses a system that would use three layers, such as "a site-specific user interface class that operates in conjunction with the user interface code and the developer user interface class," and there would be no motivation to modify either system to provide a third layer. This rejection must be REVERSED.

In regards to claims 11, 16 and 19, the Examiner adds no further substantive basis for rejection, and relies on the rejection of claims 4, 5 and 15. However, claim 19 includes "a site-specific feature class including two or more site-specific features and a corresponding site-specific state class including one or more site-specific states for each site-specific feature." There is nothing in either claim 4, 5 or 15 that is related to a state class, and that term is not used in any of claims 4, 5 or 15. As such, there is no support for the Examiner's rejection of claim 19, and it must be REVERSED.

5. **Chiang and further in view of Bickerton, in view of Wong, in view of Collins, in view of Andrews, in view of Pavela fails to provide a prima facie basis for the rejection of claims 6, 7, 13 and 14.**

In order to reject claims 6, 7, 13 and 14, the Examiner must string together six (6) unrelated references from different fields of endeavor, and has provided no motivation whatsoever for what is an obvious use of the claims as a blueprint for piecing together the prior art, using only hindsight. This is improper. *See McGinley v. Franklin Sports, Inc.*,262 F.3d 1339, 1351 (Fed. Cir. 2001) ("To prevent hindsight invalidation of patent claims, the law requires some `teaching, suggestion or reason' to combine cited references.") The only excuse provided by the Examiner for this flagrant violation of a Federal Circuit mandate is that it "would have been obvious to one of ordinary skill in the art . . . to use the various classes of the prior art with Chiang's user interface generation, in order to provide a modifiable business software generation environment." All six? Surely, if such a large combination of such a wide-ranging batch of unrelated prior art references was obvious, there would be some reference that

includes all of the various features that the Examiner must turn to six separate and unrelated references for. Presumably, the only reason provided by the Examiner is applicable to all six of the references that must be combined with Chiang, so each of those references is considered in light of that reason.

Is there any reason why one of ordinary skill in the art would read Chiang as needing features from five other references "to use the various classes of the prior art with Chiang's user interface generation, in order to provide a modifiable business software generation environment?" No. Chiang requires a team of graphic designers, business analysts and web developers to provide that function. Combination with prior art processes is not disclosed or suggested by Chiang.

Likewise, Bickerton discloses an integrated business-to-business Web commerce and business automation system that "automates to the greatest degree possible, in a unified and synergistic fashion and using best proven business practices, the various aspects of running a successful and profitable business." One of ordinary skill would not read the unified and synergistic system of Bickerton as requiring (or even permitting) anything additional, or providing any motivation for combination with other art.

Wong also fails to provide any motivation to be combined with prior art to provide a modifiable business software generation environment, and is instead oriented towards a "method and system for assisting input of information that dynamically translates information being input, and allows user interaction with the translation process. . . . Dictionaries are also supported, including optional properties such as grammar and frequency. Dictionaries may also be chained." Thus, it is apparent that Wong has nothing whatsoever to do with a modifiable business software generation environment, and instead only relates to language translation.

A review of the other cited art reveals that it, too, fails to provide any motivation to be combined to "provide a modifiable business software generation environment," and it in fact teaches away from any such combination. The Examiner has improperly relied on hindsight to jumble together six unrelated pieces of prior art, using the claims as a roadmap. These rejections must also be REVERSED.

VIII APPENDIX OF CLAIMS (37 C.F.R. 41.37(c)(8))

The text of the claims involved in the appeal are:

1. A system for generating user interface code comprising:

a user interface class system generating user interface class code, wherein the user interface class code includes two or more user interface features that can be selected and assembled into a user interface by a user; and

a handler class system generating handler class code, wherein the handler class code includes one or more states for each user interface feature of the user interface class;

wherein the user interface class and the handler class cause the selected user interface features and associated states for the user interface features to be automatically generated when the user interface code is executed.

2. The system of claim 1 wherein the user interface class system further comprises a developer user interface class system automatically generating a developer user interface class that operates in conjunction with the user interface code to provide modified user interface features.

3. The system of claim 1 further comprising a developer handler class system generating a developer handler class that operates in conjunction with the user interface code to automatically provide modified user interface states.

4. The system of claim 2 wherein the user interface class system further comprises a site-specific user interface class system generating a site-specific user interface class that operates in conjunction with the user interface code and the developer user interface class to provide site-specific user interface features.

5. The system of claim 3 wherein the developer handler class system further comprises a site-specific handler class system generating a site-specific handler class that operates in conjunction with the user interface code and the developer handler class to provide site-specific user interface states.

6. The system of claim 1 wherein the user interface class includes an accounts payable user interface class, an accounts receivable user interface class, a general ledger user interface class, a global user interface class, an inventory control user interface class, an order common user interface class, a purchase order user interface class, a report user interface class, a shipping order user interface class, a utility user interface class, a report user interface class, a test user interface class, and a template user interface class.

7. The system of claim 1 wherein the handler class includes an accounts payable handler class, an accounts receivable handler class, a general ledger handler class, a global handler class, an inventory control handler class, an order common handler class, a purchase order handler class, a report handler class, a shipping order handler class, a utility handler class, a report handler class, a test handler class, and a template handler class.

8. A method for generating user interface code comprising:

receiving a selection of user interface feature for a user interface class;

automatically retrieving a handler associated with the user interface feature that includes one or more states; and

automatically generating one or more code elements that cause a user display to be generated when executed that includes the user interface feature having the one or more states.

9. The method of claim 8 further comprising:

receiving a selection of a developer user interface feature from a developer user interface class; and

automatically generating one or more code elements that cause the user display to be generated that includes the developer user interface feature.

10. The method of claim 8 further comprising:

receiving a selection of a developer user interface state from a developer handler class; and

automatically generating one or more code elements that cause the user display to be generated that includes the developer user interface state.

11. The method of claim 9 further comprising:

receiving a selection of a site-specific user interface feature from a site-specific user interface class; and

automatically generating one or more code elements that cause the user display to be generated that includes the site-specific user interface feature.

12. The method of claim 8 further comprising:

receiving a selection of site-specific user interface state from a site-specific handler class; and

automatically generating one or more code elements that cause the user display to be generated that includes the site-specific user interface state.

13. The method of claim 8 wherein receiving a selection of the user interface feature from the user interface class includes receiving a selection from an account payable user interface class, an accounts receivable user interface class, a general ledger user interface class, a global user interface class, an inventory control user interface class, an order common user interface class, a purchase order user interface class, a report user interface class, a shipping order user interface class, a utility user interface class, a report user interface class, a test user interface class, and a template user interface class.

14. The method of claim 8 wherein retrieving the handler associated with the user interface feature that includes one or more states includes retrieving an accounts payable handler class, an accounts receivable handler class, a general ledger handler class, a global handler class, an inventory control handler class, an order common handler class, a purchase order handler class, a report handler class, a shipping order handler class, a utility handler class, a report handler class, a test handler class, and a template handler class.

15. A system for generating software code comprising:

a primary code generator receiving one or more user selections for one or more classes and generating primary software code;

a developer code generator receiving the primary software code and one or more user selections from one or more developer classes and automatically generating developer software code;

a primary code editor automatically modifying one or more of the classes; and

wherein the modifications made to the one or more of the classes by the primary code editor result in the generation of code that is backwards and forwards compatible with other tiers in a multi-tier architecture of the developer software code.

16. The system of claim 15 further comprising a site-specific code generator receiving the primary software code, the developer software code, and one or more selections from one or more site-specific classes and generating site-specific software code, wherein the modifications made to the one or more of the classes by the primary code editor result in the generation of code that is compatible with the site-specific software code.

17. The system of claim 15 wherein the primary code generator further comprises a feature class including two or more features and a corresponding state class including one or more states for each feature.

18. The system of claim 15 wherein the developer code generator further comprises a developer feature class including two or more developer features and a corresponding developer state class including one or more developer states for each developer feature.

19. The system of claim 16 wherein the site-specific code generator further comprises a site-specific feature class including two or more site-specific features and a corresponding site-specific state class including one or more site-specific states for each site-specific feature.

20. The system of claim 15 wherein the primary code generator further comprises a user interface class including two or more user interface features and a corresponding state class including one or more states for each user interface feature.

IX EVIDENCE APPENDIX (37 C.F.R. 41.37(c)(9))

This page intentionally left blank

X RELATED PROCEEDINGS APPENDIX (37 C.F.R. 41.37(c)(10))

This page intentionally left blank.

Dated: November 13, 2006

Respectfully submitted,

JACKSON WALKER L.L.P.

By: _____
Christopher J. Rourk
Reg. No. 39,348

901 Main Street, Suite 6000
Dallas, Texas 75202
Telephone: (214) 953-5990
Facsimile: (214) 661-6604